LEARNING OBSTACLE BEHAVIOR FOR IMPROVED THREAT MAPPING DURING NAVIGATION

by

Kunal Saluja

A thesis submitted to Johns Hopkins University in conformity with the requirements for the degree of Master of Science in Engineering in Robotics

Baltimore, Maryland

May, 2016

ABSTRACT

A novel obstacle evaluation system for the visually impaired is proposed and evaluated in this thesis. This system assesses the threat in the environment posed by various kinds of stationary and moving obstacles. It evaluates the behaviors of different classes of obstacles and learns features based on their collision avoidance strategies. For the purpose of crowd simulation, Reciprocal Collision Avoidance strategy has been used here. This idea can further be generalized to any multi-agent environment and can serve as the building block for new obstacle avoidance algorithms.

Primary Reader : Dr Austin Reiter

DECLARATION OF AUTHORSHIP

I, Kunal Saluja, declare that this thesis titled, 'Learning Obstacle Behavior For Improved Threat Mapping During Navigation' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Master's degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly at-ributed.
- Where I have quoted from the work of others, the source is always given.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Austin Reiter for his continuous support, patience and immense knowledge. His guidance helped me in times of research and writing of this thesis.

Besides my advisor, I would like to thank Mr. Kapil Katyal and the team of HARMONIE at Johns Hopkins University Applied Physics Laboratory, for their continued support during this project. Without their expert comments and technological support , this thesis wouldn't have been possible.

I would also like to thank my family: my aunt Anita Saluja, for enabling me to pursue my dreams at Johns Hopkins. Without her support, none of this would be possible.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Over the past 20 years, the number of visual impairments caused due to infectious diseases has significantly reduced. However, the risk of age-related visual impairment has been on a steady rise both in developing and developed countries . In terms of numbers, 285 million people are estimated to be visually impaired worldwide, out of which 39 million are blind and 246 million have low vision [1].

Visual impairment hinders a person's daily activities, the most affected being their ability to travel freely. Low range vision limits their ability to estimate the speed and trajectory of obstacles in the environment.

There are two essential parts of Human way finding – detection of obstacles in the environment and navigating around it (Golledge et al., 1991; Golledge, Klatzky, and Loomis, 1996; Rieser, Guth, and Hill, 1982; Strelow, 1985; Welsh and Blasch, 1980). The major challenge is to tackle the lack of critical information needed for bypassing obstacles - self velocity, heading and important features in the immediate environment- information that all sighted individuals use for navigating through familiar or unfamiliar conditions.

The visually impaired mostly rely on a cane for navigation in unfamiliar environments. But due to the recent advances in technology, digital cameras and laser systems are starting to show their importance in this specific scenario. Recent efforts suggest a trend towards developing reliable computer vision systems for navigation, as cameras are relatively inexpensive and compact than most laser based systems.

This thesis aims to be a building block for developing more advanced navigation systems. We have tried to assess the immediate environment of a visually impaired person by creating a “detection pipeline”- which detects the obstacles, identifies its class (human, table or a bike) and assigns a threat value to every obstacle. Using this threat value, we create a threat heat map of the surrounding environment, which can be used by any global navigation algorithm.

The distinguishing aspect of this thesis is the proposal of a methodology using which, this system can analyze the trajectory of obstacles around the subject and adjust its estimate of the threat posed by the obstacle. We use a number of critical features to categorize an object's trajectory which helps us to evaluate the threat posed by it.

The rest of this thesis consists of 4 chapters. Chapter 2 summarizes the past research in this field, while Chapter 3 explains the entire system design. We present the simulation framework in Chapter 4 , followed by the results and conclusion in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

Many Navigation techniques utilize the Global Positioning system to move from one point to another. Loomis was one of the first to use GPS with an FM correction data receiver for an accurate determination of the location of the traveler[5]. A notably similar approach is taken by MoBic [6] and Hideo Makino [7] et al.

Some more systems using GPS are the BrailleNote GPS from the Sendero Group [8] and Bruce Thomas, etc [9]. BrailleNote GPS is commercially available and provides the subject with distance to destination and nearby location names . However, one of the biggest limitation of a GPS guided system is that it is only available outdoors. Therefore it is not very useful for local path planning and collision avoidance.

Sunita Ram and Jennie Sharf [2] designed the "People sensor," which uses pyroelectric(thermal sensing element) sensor to distinguish between human and non human obstacles, while using an ultrasonic sensor to measure the distance to these obstacles. It aims at reducing the possibility of embarrassment by helping the subject to avoid unintended cane contact with other pedestrians.

John Zelek [3] developed a hand wearable which provides tactile feedback about the surrounding environment. It uses two web-cams to detect the obstacles and the tactile information is relayed to the subject using a vibrating glove. The vibrating buzzers in the fingers of the gloves send impulses to the user, warning about the terrain irregularities up to 30 feet ahead.

Metronaut [4], which uses a bar code reader, was developed by Asim Smailagic and Richard Martin. It is a novel wearable computer system that calculates position from a series of bar code labels placed at strategic locations at CMU's campus. Similarly, A. R. Golding and N. Lesh [10] used 3D accelerometer, a 3D magnetometer, a fluorescent light detector and a temperature sensor to predict the user's current information.

Over the past few years, computer vision techniques have started being used to determine the position and velocity of the subject. Stereo cameras carry huge significance as they can express the depth of the objects in the field of view. The Optophone [11] uses the edge detection scheme to calculate the depth map. The depth map is then converted into sound where the loudness of sound is directly proportional to the intensity of the pixel.

An important work in this area is the system developed by Yoshihiro Kawai and Fumiaki Tomita [12]. This system has a computer, a sound processor, three small cameras and a headset. The object recognition is performed on the 3D data obtained and it is then converted to sound, though the primary purpose of this system is not to aid in navigation.

Our system is designed to identify the obstacles in the vicinity and calculate the relative threat posed by it based on its behavior around the user. For example, it is less dangerous for a visually impaired person to walk towards a group of humans as opposed to a static wall. The humans are likely to avoid the collision and are therefore perceived as having lower threat by the user.

CHAPTER 3

SYSTEM DESIGN

The following diagram depicts the system pipeline

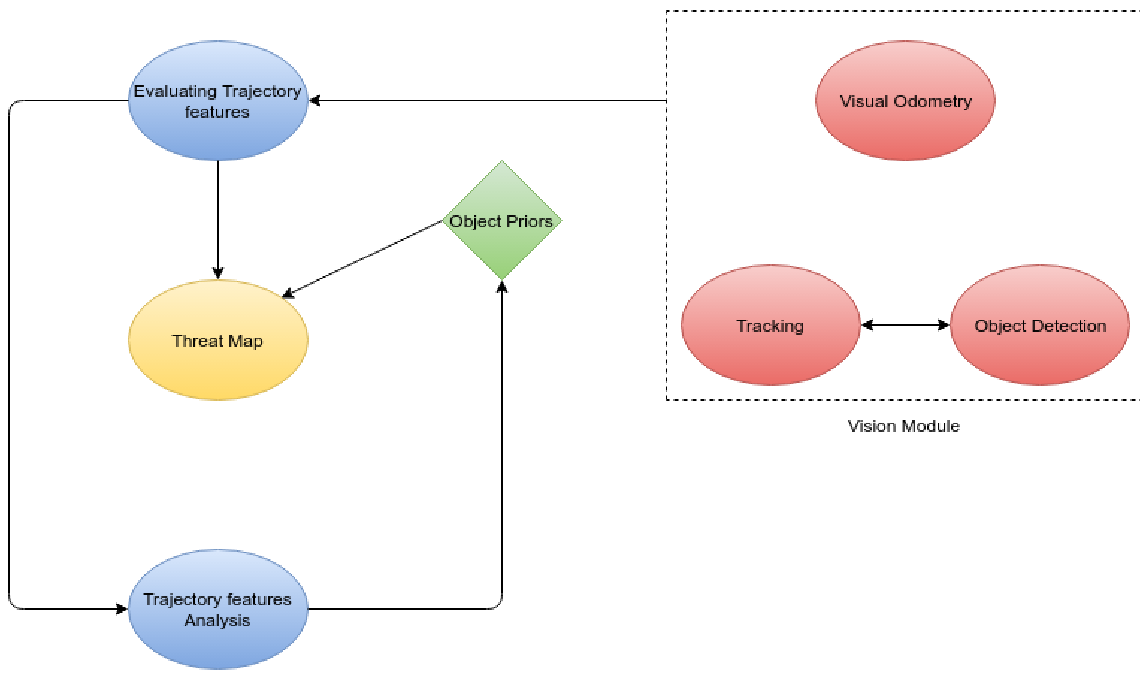


Figure 1: System design

The figure above shows the design pipeline of the system. The vision module is responsible for calculating the position, velocity and the trajectory of the user and the obstacles in the field of view. Once this information is obtained, "trajectory features" are calculated for every obstacle, which form the basis of the threat evaluation system. The variation in the obstacle avoidance methodologies of different obstacles is captured by these trajectory features. The "object prior", which is a unique value between 0-1 assigned to every obstacle, is the entity learned on the fly. It reflects our assumption of how dangerous any obstacle is, and its value is constantly reinforced using the threat values.

A threat map is then created around the user which is updated at every time step.

We will now explain every component in the design pipeline.

3.1 Visual Odometry

Visual odometry is the process of determining the position and orientation of an autonomous agent using associated images. The basic algorithm is as follows:

- Images are undistorted
- Harris features are detected on each image. Features are correlated across the corresponding images and outliers are removed
- 3D points are generated by triangulation
- Points are collected over a certain number of frames. Then the 3 point algorithm is used to calculate the Rotation and Translation between successive frames. Reprojection is done on both left and right images for scoring and iterative refinement
- This process is repeated till the re projection error is reduced

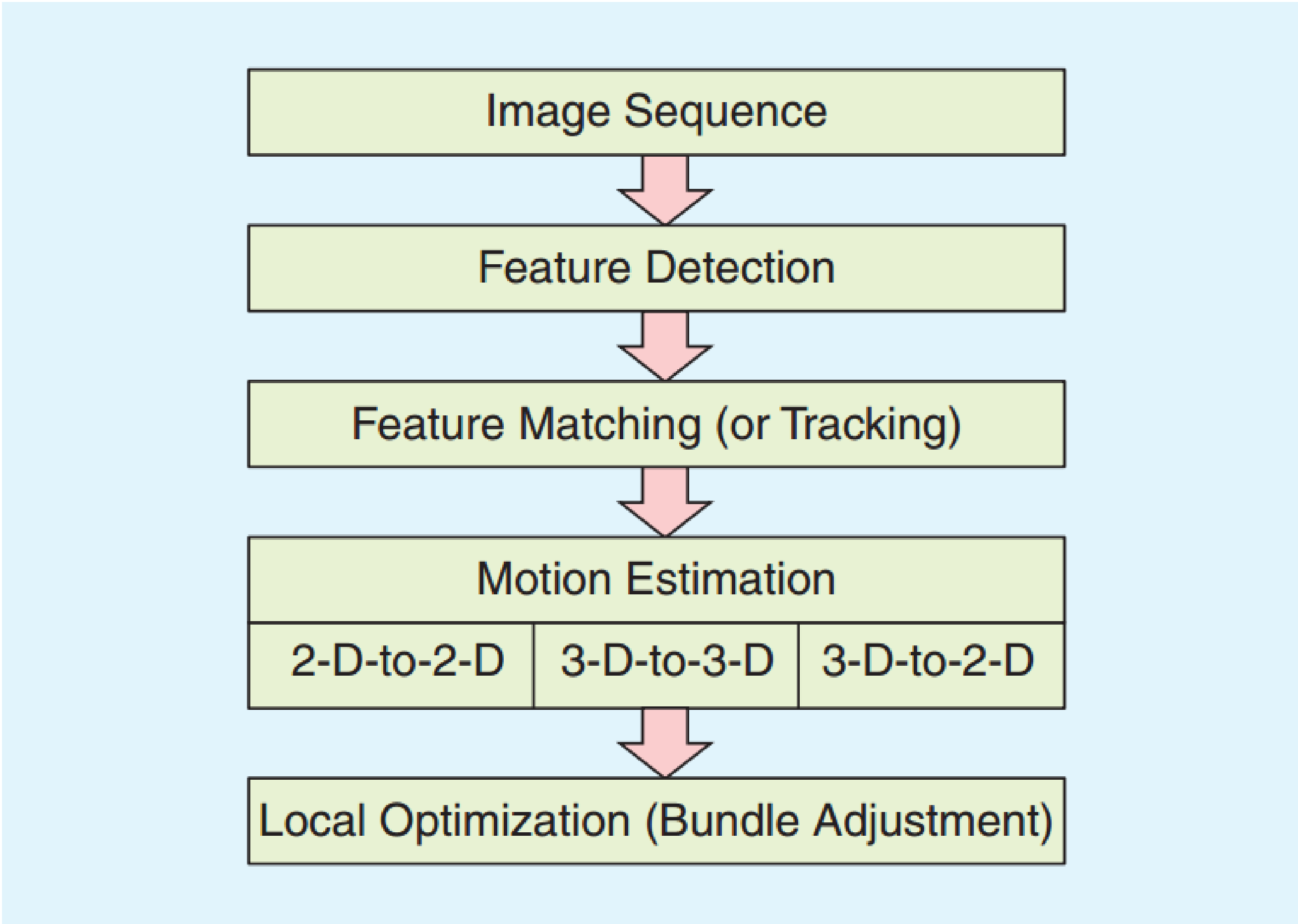


Figure 2 : Visual odometry

3.2 Object Recognition

The prime idea of this thesis is to allow the threat values to adapt to different classes of obstacles in the user's local environment. The first step is to recognize the different kinds of obstacles in the environment. This can be achieved using state of the art object detectors, which are not a hindrance to the real time requirements of the system.

Recent advances in convolutional neural networks (CNN) have resulted in a significant jump in object detection standards . It is now clear that CNN based features outperform the traditional SIFT/HOG features.

Using sliding window detectors with a CNN is a slow process, unsuited for real time applications. Therefore we aim to use the region based CNN, which involves generating category independent region proposals, as it is a much faster approach. A typical R-CNN detector consists of 3 modules:

1. Independent region proposal generator
2. The actual CNN which generates a feature vector for each region
3. A set of class specific SVMs

The Pycaffe implementation of faster R-CNN, trained specifically for the most commonly encountered obstacles should do the job for us.

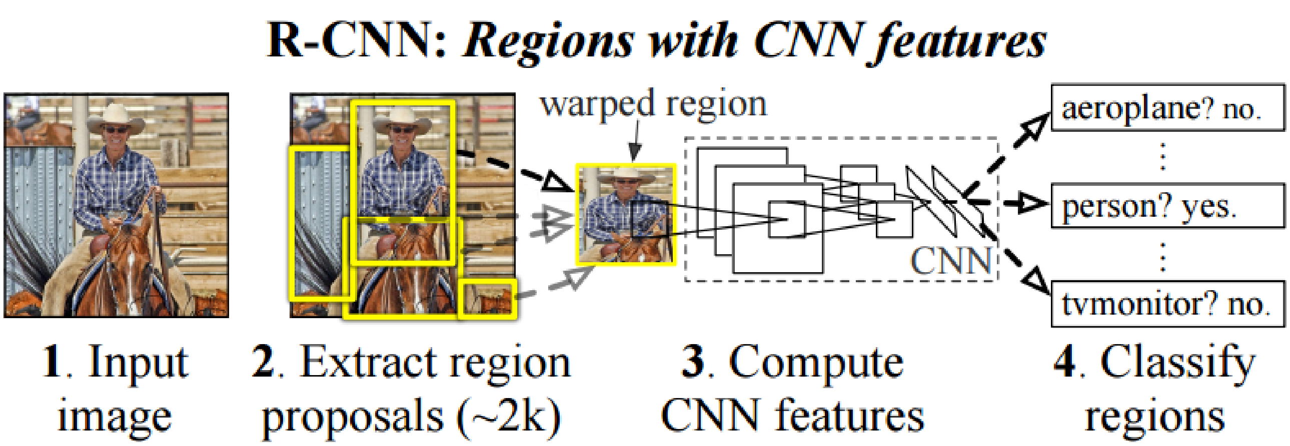


Figure 3: R-CNN framework[17]

3.2.1 Suggested Improvisations

Though the R-CNN and its faster variants (Faster R-CNN) are fast and accurate, we can further increase the detection rate if we limit the scope of our detection. We can assume that we are looking for a fixed classes of obstacles and train the region proposal generator accordingly. There are a number of region proposal techniques. The ranking method proposed by Zitnick and Dollar (Edge boxes) shows high efficiency and recall.

Since we are required to track the objects to obtain their trajectories, the tracking algorithm is bound to follow next in the system pipeline. After obtaining a highly confident detection, we propose to generate region proposals within a small region around the detection. This will not only increase the detection speed, but will also increase the true positive rate. Coupling the object detection with the tracking algorithm can increase the effectiveness of this approach in the context of our needs.

3.3 Object Tracking

In order to keep the processing time for the entire pipeline, we aim to use the Continuous Adaptive Mean Shift (Camshift) Algorithm. As stated in the above section, we aim to couple this approach with the object detection scheme. The basic outline of the algorithm is as follows:

- Run the Faster R-CNN over the left and right image of the stereo camera.

Preserve common detections which have a high confidence factor.

- Run a separate tracker for every new detection in a new thread. Set the initial location of the Mean shift search window as the bounding box for the detections above.
- Calculate the color histogram for each ROI
- Iterate Mean shift algorithm to find the centroid, then store the distribution area and centroid location
- For the next frame, center the search window at the previously found mean location and set the window size to any function of the distribution area previously stored. Repeat from point 3.
- Once the iterations have converged, we have a location in the left image for the object. We can calculate the mean depth value for these set of pixels by projecting them into space and that should serve as the tracking point in space. We can verify this projection by back projecting it on the right image.

3.4 Trajectory Features

Every class of obstacles presents a different behavior when it encounters the visually impaired user. For example, humans will tent to navigate around differently as opposed to a stationary wall (which will not avoid at all), or a bike. We aim to capture this difference in obstacle avoidance behavior and convey this information in a intuitive way to the user.

The features being used currently are:

1) Closest distance attained by the obstacle while it is within the field of view of the user

2) Average speed of the obstacles computed via visual odometry

3) Measure of responsiveness – This feature measures the maximum deviation recorded in an obstacle's trajectory in the vicinity of the user. It is reflective of the degree of holonomicity of the obstacle. As we can see in the figure below, Angle 1 is higher as compared to Angle 2, because the response of the obstacle (which traced the red trajectory) was slower as opposed to the response of the agent which traced the green trajectory. This reflects that latter is a better planner.

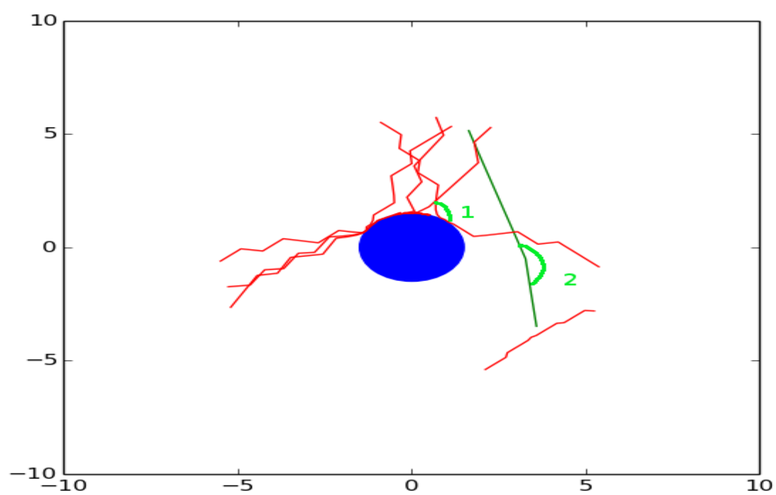


Figure 4: Measure of responsiveness

3.5 Threat values

We intend to convey the gathered information to the agent by presenting the immediate environment as a threat map. Each obstacle is assigned a threat value ranging from 0-1.

First let us define some variables:

$$\vec{V}_R = Relative\ velocity\ of\ obstacle\ w.r.t\ user$$

$$\vec{V}_A = Velocity\ of\ user$$

The threat is calculated using a heuristic as follows:

$$Threat = Prior \times \left(\frac{\vec{V}_A \cdot \vec{V}_R}{|Max\ velocity\ of\ obstacle| + |max\ velocity\ of\ agent|}\right) \times \left(\frac{2 \times agent\ radius}{distance\ between\ obstacle \wedge user}\right)$$

For creating the threat plot of the environment, **every point** in the mesh grid representing the immediate surrounding of the subject calculates the following value:

$$Mesh\ point_{xy} = \sum_{obstacle=obstacle\ 1}^{obstacle\ n} \frac{Threat_{obstacle}}{Distance_{obstacle \wedge user}}$$

The values are then normalized to be displayed on the map and a suitable colormap is chosen.

CHAPTER 4

SIMULATION FRAMEWORK

The biggest challenge is to correctly simulate the behavior of walking humans and other obstacles, which a visually impaired user might encounter in every day life. We tried to model this behavior by adapting the Optimal Reciprocal Velocity Obstacle algorithm, which has been used significantly in the academic community for crowd simulation. In the following section, we explain how this concept has evolved over the years.

4.1 Reciprocal Velocity Obstacles

This algorithm is intended to provide real-time multi agent navigation where each agent has no explicit communication with each other. This concept is an extension of the "Velocity Obstacle" [13] which we will briefly describe here.

4.1.1 Velocity Obstacles

Let the agent A be positioned at P_A and the agent B at position P_B. The velocity obstacle($VO^A{}_B(v_B)$) of agent B w.r.t agent A is the set of all velocities of agent A which will result in a collision at any moment of time, if agent B has the velocity v_B.

Some important properties of Velocity obstacles are as follows:

1) Symmetry : If $v_A \in VO^A{}_B(v_B)$, then $v_B \in VO^B{}_A(v_A)$

2) Translation Invariance: $v_A \in VO^A_B(v_B) \leftrightarrow v_A + u \in VO^A_B(v_B + u)$

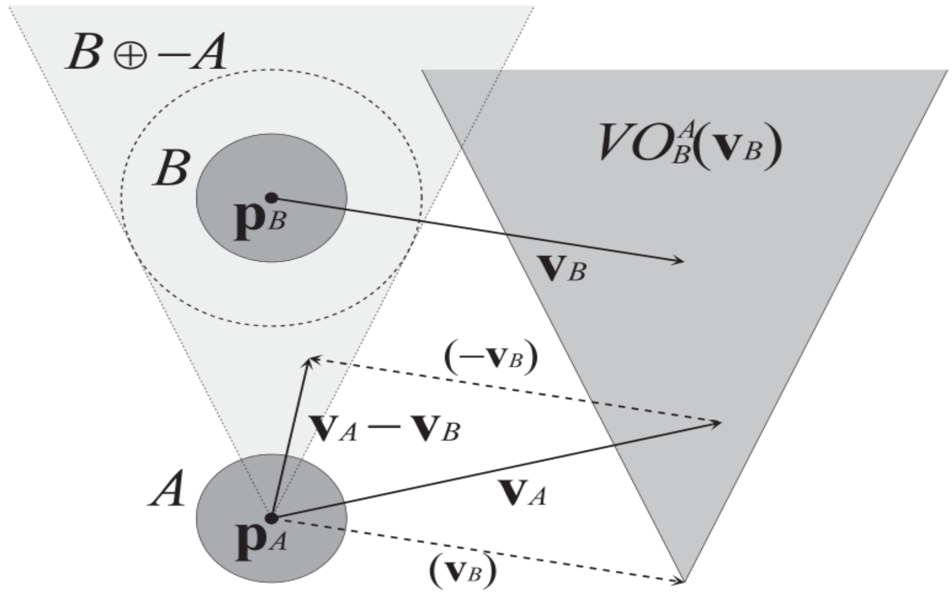


Figure 5: The Velocity Obstacle $VO^A_B(v_B)$ a disc-shaped agent A [14]

One of the biggest disadvantages of velocity obstacles is its inherent oscillation. Suppose two agents are moving with velocities V_A and V_B respectively and at some moment $V_A \in VO^A_B(V_B)$ and $V_B \in VO^B_A(v_A)$. Now they choose velocities $V_{A'}$ and $V_{B'}$ such that they do not lie in the velocity obstacles of each other. If in this new scenario, previous velocities (V_A and V_B) lie outside the current velocity obstacle of each agent, they will be selected again, thus leading to oscillations.

4.1.2 Extension to Reciprocal Velocity Obstacles

Instead of choosing a new velocity outside the agent's velocity obstacle, this concept simply chooses the average of

i) A velocity outside the velocity obstacle

ii) Current velocity of the agent.

There are certain guarantees associated with this concept namely the a) Collision free navigation and b) Oscillation free nature.[14]

4.1.3 Optimal Reciprocal Collision Avoidance

By applying the above concepts to multi agent problems, the authors of the Reciprocal velocity obstacle technique came up with this collision avoidance strategy[15]. For a given time Γ, velocities of agents A and B are selected outside of each others velocity obstacles such that the following conditions are met:

1) They avoid collisions for atleast Γ seconds

2) Since there are many sets of such velocities, a pair of velocities is selected such that it as close to the optimal velocity of each agent.

The basic framework is explained in this figure 5

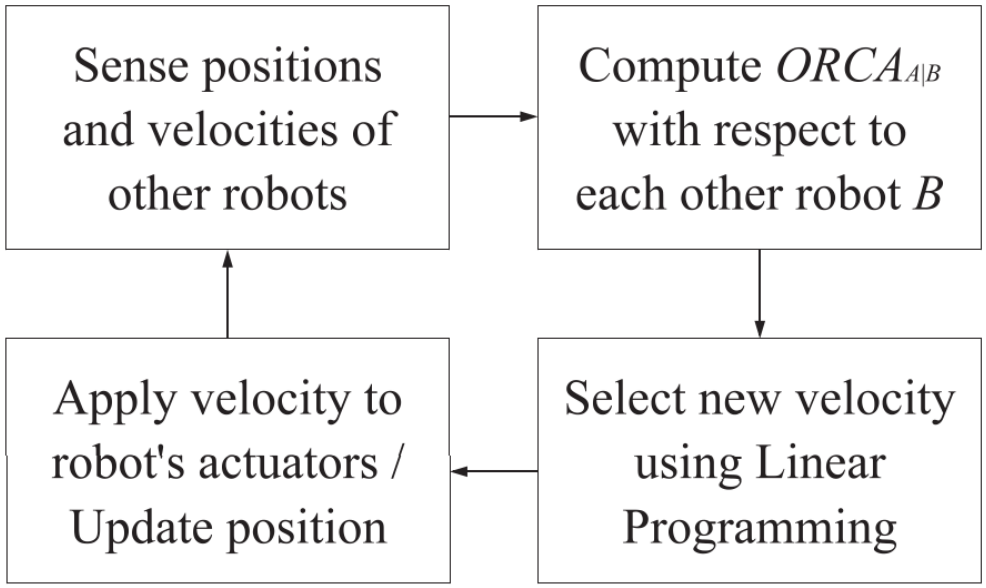


Figure 5: A schematic overview of the continuous cycle of sensing and acting that is independently executed by each robot[15]

4.2 Setup

The simulation consists of a certain number of user specified agents in a field, who are attempting to reach a goal point. The field consists of a number of circular obstacles as well. Each agent is attempting to reach a goal point and once it reaches the goal point, it is assigned a new goal point. We use the RVO2 Library[16], which was been used significantly in multi agent simulation in academia. For every agent we add to the simulation, we have to control the following parameters:

Factors	**Explanation**
Position	The two-dimensional starting position of this agent.
Neighbor distance	The maximal distance (center point to center point) to other agents this agent takes into account in the navigation. The larger this number, the longer the running time of the simulation. If the number is too low, the simulation will not be safe. Must be non-negative.
Max neighbors	The maximal number of other agents this agent takes into account in the navigation. The larger this number, the longer the running time of the simulation. If the number is too low, the simulation will not be safe.
Time horizon	The minimal amount of time for which this agent's velocities that are computed by the simulation are safe with respect to other agents. The larger this number, the sooner this agent will respond to the presence of other agents, but the less freedom this agent has in choosing its velocities. Must be positive.
Time horizon obstacle	The minimal amount of time for which this agent's velocities that are computed by the simulation are safe with respect to obstacles. The larger this number, the sooner this agent will respond to the presence of obstacles, but the less freedom this agent has in choosing its velocities. Must be positive.
Radius	The radius of this agent. Must be non-negative.
Max speed	The maximal speed of this agent. Must be non-negative.
Velocity	The initial two-dimensional linear velocity of this agent (optional).

Table 1: Factors involved in simulation[16]

We also define a “learning radius” value, which reflects the range of the stereo camera mounted on the user. For simplicity, all agents are assumed to have a circular field of view.

Two simulation frameworks were prepared.

1) Python based simulation : This simulation is aimed at learning the trajectory features and trying different learning approaches. The threat map is created for the visually impaired agent as it moves in the field.

2) Gazebo Ros Simulation : This simulation extends the general concept of this thesis to multi agent robots, where every agent is a turtle-bot. This was aimed at creating the entire data pipeline as proposed in Chapter 3, and testing it on the robots.

CHAPTER 5

RESULTS, DISCUSSION AND CONCLUSION

5.1 Simulation Parameters

The global parameters define the overall characteristics of the simulation.

Global Parameters	Value
Number of agents	25
Number of classes	5
Number of obstacles	7
Agent radius	1.5

Table 2: Simulation parameters

We simulate 5 different classes of agents in the simulation, in addition to the single visually impaired agent. Here are the parameters of each class and the test subject.

	Class 1	Class 2	Class 3	Class 4	Class 5	Test subject
Neighbor distance	15	15	15	15	15	4.5
Max neighbors	2	3	4	5	6	4
Time Horizon	0.3	0.9	1.8	2.7	4.5	0.6
Time Horizon for Obstacles	0.9	0.9	0.9	0.9	0.9	0.3
Preferred velocity	5	5	5	5	5	4
Max velocity	7	7	7	7	7	6

Table 3: Class variation in simulation parameters

Learning Radius for the visually impaired agent is set to be 6.

5.2 Explanation for Choice of variables

The motivation behind using agents of varying classes is to enable the visually impaired agent to identify the distinct features without prior knowledge. We vary the "time horizon" values from class 1 to class 5 such that the agents become better planners i.e. the agents with higher value plan for future collisions robustly.

The "Max neighbor" value also increase from class 1 to class 5. This effectively means that the agent with a higher value will consider more neighbors while planning for future collisions.

Adding the effects of these two parameters, we can say that the "**agents become smarter"** as we move from class 1 to class 5.

The Visually impaired agent on the other hand has a slower speed than other agents and plans ahead for lower time due to limited visual information. Hence its parameters have been scaled down proportionally.

The preferred speed for all classes is intentionally kept same to identify distinguishing features apart from speed.

5.3 Threat Maps

The following series of figures shows the transition of threat map as a moving obstacle moves past the visually impaired agent:

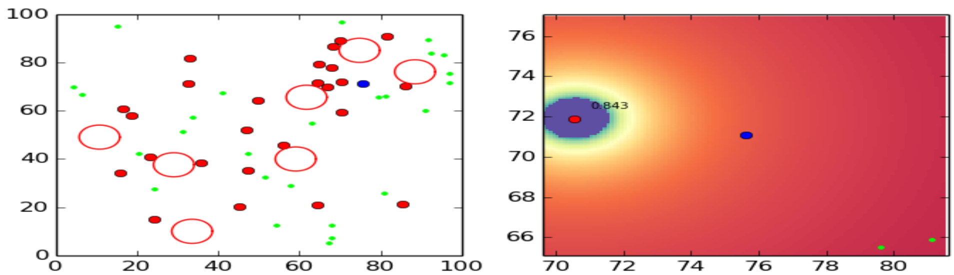


Figure 6(a)

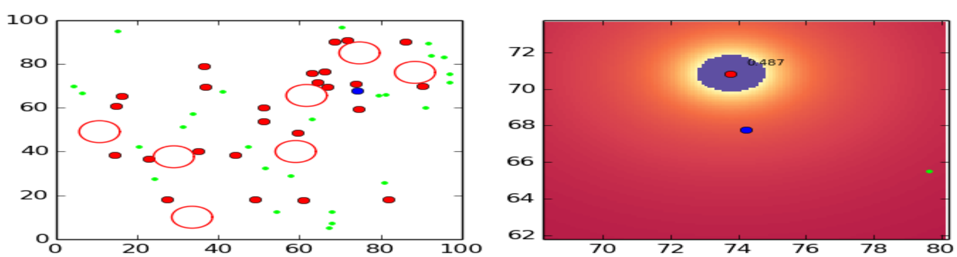


Figure 6(b)

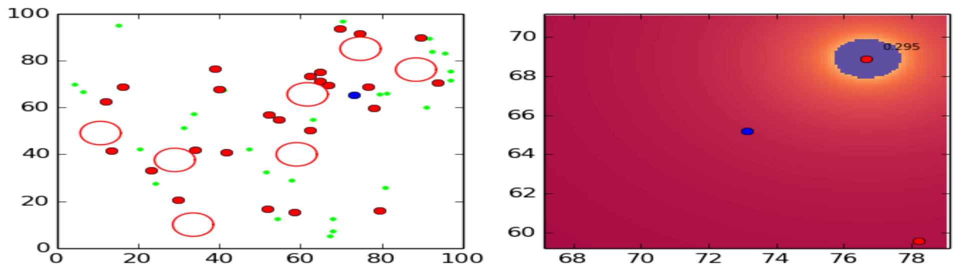


Figure 6(c): Threat maps shown in (a),(b) and (c)

The blue colored circles represent the visually impaired agent as it navigates in the field. The red dots represent the moving obstacles. All agents are assigned a random goal point , represented by the green dot. The obstacles are denoted as circles on the field.

On the right half, the corresponding threat map is displayed around the visually impaired agent. It shows how the threat in the environment changes as the moving agent passes by the subject. The threat values fall progressively, so do the threat levels in the surrounding environment.

We use the “spectral” colormap to reflect the threat values (which are normalized).

Figure 7: Spectral Colormap

5.4 Trajectory features

Since our goal is to calculate a prior value for each type of obstacle, it is important to study how dangerous the obstacle is while it is moving around the visually impaired agent. We aim to distinguish between these obstacle on the basis of the trajectory features as defined in section 3.4.

We tried to capture the variation of these features across the different classes of obstacles over multiple simulations. Here are some results:

a) **Closest distance**

The following agent

	Class 1	Class 2	Class 3	Class 4	Class 5	Stationary obstacle
Average Closest Distance – Run 1	3.72	3.82	3.86	3.92	4.02	2.38
Average Closest Distance – Run 2	3.65	3.71	3.72	3.94	4.24	2.95

Table 4: Closest distances

We can clearly observe that the closest distances increase as the agent gets smarter I.E. from class 1 to 5. This is intuitively correct as the agent can plan better and hence it tries to avoid the visually impaired agent before getting too close.

b) **Measure of responsiveness (in degrees) :**

We can clearly observe that the feature value reduces as the "agent gets smarter" I.E as the agent becomes a better planner, it can foresee the obstacle in its path early. This allows it to avoid the obstacle smoothly in smaller increments of angular deviations as compared to other "not so good" planners who need to make sudden turns to avoid the obstacle.

	Class 1	Class 2	Class 3	Class 4	Class 5	Stationary obstacle
Measure of responsiveness – Run 1	76.72	45.17	39.23	35.54	4.02	0
Measure of responsiveness – Run 2	71.34	51.26	49.12	39.60	10.12	0

Table 5: Measure of responsiveness

c) Average Speed

Since we are obtaining results from simulations, we already possess the information about all the agents. Therefore we do not perform any experiments related to this feature. However, on an actual system, this feature is bound to play a critical role as all agents usually have a characteristic speed, which can help distinguish between them.

Based on these observations and general intuition, we can propose that the prior which we aim to learn for every type of obstacle has the following properties:

$Prior \propto 1/Closest\,distance$: The closer the obstacles get to the visually impaired agent, the more dangerous they are

$Prior \propto Measure\,of\,responsiveness$: Since measure of responsiveness indicates how good an agent plans its path, we can say that the prior value is higher for agents which react which higher angular deviations.

$Prior \propto Average\,Speed$: Agents with higher speed should pose more threat to the visually impaired agent.

5.5 Conclusion and Future work

In this thesis, we have proposed a system which will aid visually impaired people in recognizing the threats in the environment. The trajectory features indicate a clear distinction between different type of obstacles. This enables us to learn priors for different classes of obstacles and treat the threat associated with them differently.

There are a number of challenges left to overcome:

a) Discovering advanced heuristics for threat mapping in accordance with the correct understanding of the environment.

b) Implementing the system design pipeline

c) Improving the learning framework

The threat map distribution has potential applications in the development of advanced obstacle avoidance algorithms aimed specifically at guiding the visually impaired people through rough terrains.

BIBLIOGRAPHY

[1] World Health Organization, Webpage,

http://www. who.intlmediacentre/factsheets/fs282/en/, last visited on April 19th, 2016

[2]Ram, S.; Sharf, J., "The People Sensor: A Mobility Aid for the Visually Impaired." Second International Symposium on Wearable Computers, Digest of Papers, 1998, Page(s): 166 –167

[3]Zelek, J., 2002. "The E. (Ben) & Mary Hochhausen Fund for Research in Adaptive Technology For Blind and Visually Impaired Persons."

[4]A. Smailagic and R. Martin, "Metronaut: A Wearable Computer with Sensing and Global Communication Capabilities", The First International Symposium on Wearable computer, Boston MA, 1997, pp. 116-122

[5]Loomis, J.M.; Golledge, R.G.; Klatzky, R.L.; Speigle, J.M. & Tietz, J., "Personal Guidance System for the Visually Impaired." Proceedings of the 1st Annual ACM/SIGGAPH Conference On Assistive Technology Page(s): 85-91, 1994

[6]Strothotte, T., Petrie, H., Johnson V. and Reichert L., "MoBIC: user needs and preliminary design for a mobility aid for blind and elderly travelers," Available from URL: http://isgwww.cs.unimagdeburg.de/projects/mobic/tidefull.html Site last seen Aug 2003

[7]Makino, H.; Ishii, I.; Nakashizuka, "Development of Navigation System for the Blind Using GPS and Mobile Phone Combination." Engineering in Medicine and Biology Society, 1996. Bridging Disciplines for Biomedicine, 18th Annual International Conference of the IEEE, Volume: 2, 1997, Page(s): 506 –507

[8]Sendero Group Inc, "Sendero Group Shopping Cart,"

[9]B. Thomas, V. Demczuk, W. Piekarski, D. Hepworth, and B. Gunther, "Wearable Computer System with Augmented Reality to Support Terrestrial Navigation," Proc. 2nd Int. Symp. On

Wearable Computers, 1998, Page(s):168-171

[10]Golding, A. R. and Lesh, N., " Indoor Navigation Using a Diverse Set of Cheap, Wearable Sensors." The Third International Symposium on Wearable Computer, Digest of Papers, 1999, Page(s): 29-36

[11]M. Capp and P. Picton, "The Optophone: an electronic blind aid", Engineering Science and Education Journal, 2000, pp. 137-143.

[12]Y. Kawai and F. Tomita, "A support system for visually impaired persons to understand threedimensional visual information using acoustic interface", IEEE 16th conference on Pattern Recognition, 2002, pp. 974 – 977

[13]P. Fiorini and Z. Shiller, "Motion planning in dynamic environments
using velocity obstacles," Int. Journal of Robotics Research, vol. 17,
no. 7, pp. 760–772, 1998

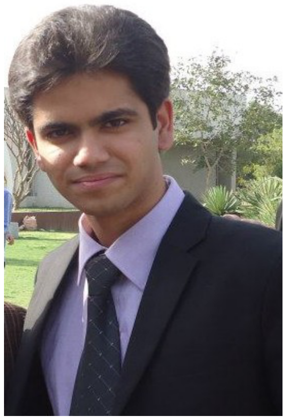
[14]Van den Berg, Jur, Ming Lin, and Dinesh Manocha. "Reciprocal velocity obstacles for real-time multi-agent navigation." Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, 2008.

[15] Van Den Berg, Jur, et al. "Reciprocal n-body collision avoidance." Robotics research. Springer Berlin Heidelberg, 2011. 3-19.

[16]Jur van den Berg, Stephen J. Guy, Jamie Snape, Ming C. Lin and Dinesh Manocha, RVO2 Library, downloaded from http://gamma.cs.unc.edu/RVO2/

[17] Ross Girshick, Fast R-CNN, Proceedings of the IEEE International Conference on Computer Vision, 1440

Vita



Kunal Saluja obtained his B.E. (Honors) degree in Mechanical Engineering from BITS Pilani in 2014. His primary research interests lie in the field of computer vision, machine learning and robotics.

He was awarded the DAAD WISE Scholarship in 2013 to pursue research in Technical University in Munich. In 2014, he was awarded the Narotam Sekhsaria Scholarship to pursue his master's degree at Johns Hopkins University. Kunal has worked with various robotics start-ups including Grey Orange Robotics and Bossa Nova Robotics. He also served as the Robotics representative in the Graduate Committee at Whiting School of Engineering (2015-2016). He will graduate with a Master's degree in Robotics (Perception and Cognition Track) from Johns Hopkins University in May 2016.